

Relatório de Laboratório 2

Circuitos Lógicos

Aluna: Rebeca de Macêdo Ferreira.

Mat.: 2016 000 524

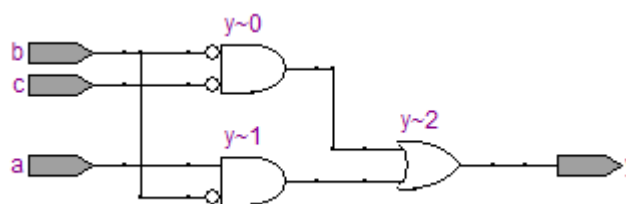
SILLYFUNCTION SIMPLIFICADA

Procedimento 1

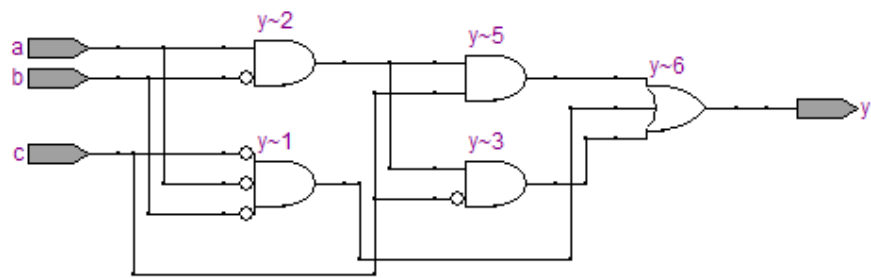
Fizemos um arquivo .vhd, agora com a função simplificada da letra B da preparação e colocamos o arquivo na pasta sillyfunction do nosso novo projeto.

```
1 library IEEE; use IEEE.STD_LOGIC_1164.all;
2
3 entity sillyfunction is
4     port(a, b, c: in STD_LOGIC;
5           y: out STD_LOGIC);
6 end;
7
8 architecture synth of sillyfunction is
9     begin
10    y <= (not b and not c) or
11         (a and not b);
12 end;
```

Em seguida compilamos o código no Quartus II e verificamos se havia erros para corrigir, então selecionamos: Tools → Netlist Viewers → RTL Viewer para termos a visualização do circuito da expressão simplificada (Fig. 1) e comparamos com o circuito da aula anterior (Fig. 2)



Circuito da expressão simplificada



Circuito da expressão não simplificada

Percebemos que o circuito da expressão simplificada é menor e consequentemente mais rápido que o de antes.

Procedimento 2

Criamos num editor de texto um arquivo (com extensão .tv) de leitura para a simulação da tabela verdade do circuito produzido.

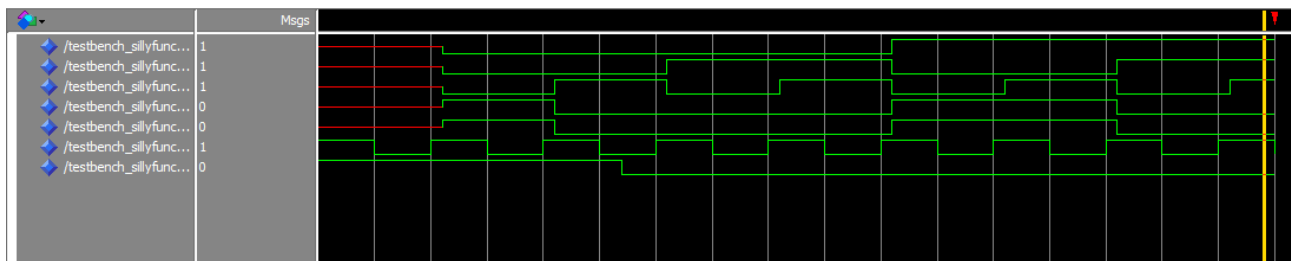
```
000_1
001_0
010_0
011_0
100_1
101_1
110_0
111_0
```

Tabela verdade do circuito produzido.

Depois fizemos um arquivo .vhd com o código de test bench. Configuramos o nome do test bench com o nome do arquivo .vhd (tomando cuidado pra deixar os nomes identicos) em: botão direito Project → Settings → EDA Tools Settings → Simulation.

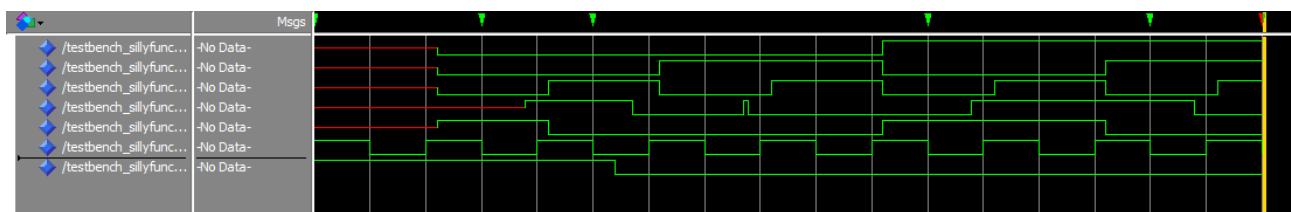
Colocamos no módulo de alto nível no test bench como testbench_sillyfunction (ou testbench_[nome do seu arquivo .vhd]). Nome da instancia: 'dut'. O arquivo .vhd do testbench deve estar numa subpasta do projeto.

Para simulação do RTL level, o arquivo .tv deve estar dentro da pasta de simulação no modelsim (verificar em Project → Settings → EDA Tools Settings → Simulation). Após isso, selecionamos Tools → Run EDA Simulation Tool → RTL Level e vemos o resultado obtido



Simulação RTL level

Simulando RTL level vemos que não tínhamos nenhum erro, porém quando simulamos o Gate level percebemos que continham alguns erros devido ao tempo de atraso das portas:



Simulação Gate level com erros (indicados pelas setas verdes acima das ondas)

```

Transcript
# view structure
# .main_pane.structure.interior.cs.body.struct
# view signals
# .main_pane.objects.interior.cs.body.tree
# run -all
** Note: Error: y = 'X'
# Time: 15 ns Iteration: 1 Instance: /testbench_sillyfunction
** Note: Error: y = '1'
# Time: 25 ns Iteration: 1 Instance: /testbench_sillyfunction
** Note: Error: y = '0'
# Time: 55 ns Iteration: 1 Instance: /testbench_sillyfunction
** Note: Error: y = '1'
# Time: 75 ns Iteration: 1 Instance: /testbench_sillyfunction
** Failure: 8 tests completed, errors =4
# Time: 85 ns Iteration: 1 Process: /testbench_sillyfunction/line_27 File: D:/CIRCUITOS/Lab2Circuitos/sillyfunction/./testbench/testbench_sillyfunction.vhd
# Break in Process line_27 at D:/CIRCUITOS/Lab2Circuitos/sillyfunction/./testbench/testbench_sillyfunction.vhd line 64
# Simulation Breakpoint: Break in Process line_27 at D:/CIRCUITOS/Lab2Circuitos/sillyfunction/./testbench/testbench_sillyfunction.vhd line 64

```

Transcript informando erros

Então alteramos no testbench_sillyfunction o tempo de clock pra ficar com os atrasos compatíveis.

```

17 process begin
18   clk <= '1'; wait for 10 ns;
19   clk <= '0'; wait for 5 ns;
20 end process;

```

Testbench_sillyfunction alterado de 5 ns para 10 ns no clock 1

Voltamos a simular o Gate level, agora com o .vhd alterado na tentativa de correção de erro, verificamos que o erro foi corrigido:



Simulação do Gate level após modificação do clock.

O RTL level considera apenas a expressão booleana, o Gate level considera as portas que a expressão usa e portas lógicas tem atrasos, por esse motivo o RTL level não deu erros e o Gate level deu.

Por fim, para fazer a visualização em formato de lista selecionamos Add → To List → Selected Signals. Obtivemos como resultado:

List - Default							
ps		/testbench_sillyfunction/a					
delta		/testbench_sillyfunction/b					
		/testbench_sillyfunction/c					
		/testbench_sillyfunction/y					
		testbench_sillyfunction/y_expected					
0	+0	U	U	U	U	U	U
0	+1	U	U	U	U	U	1
0	+2	U	U	U	X	U	1
0	+4	U	U	U	U	U	1
2591	+1	U	U	U	X	U	1
10000	+1	U	U	U	X	U	0
15000	+1	U	U	U	X	U	1
16000	+0	0	0	0	X	1	1
23882	+1	0	0	0	1	1	1
25000	+1	0	0	0	1	1	0
30000	+1	0	0	0	1	1	1
31000	+0	0	0	1	1	0	1
38548	+1	0	0	1	0	0	1
40000	+1	0	0	1	0	0	0
45000	+1	0	0	1	0	0	1
46000	+0	0	1	0	0	0	1

Visualização em lista.

GATES

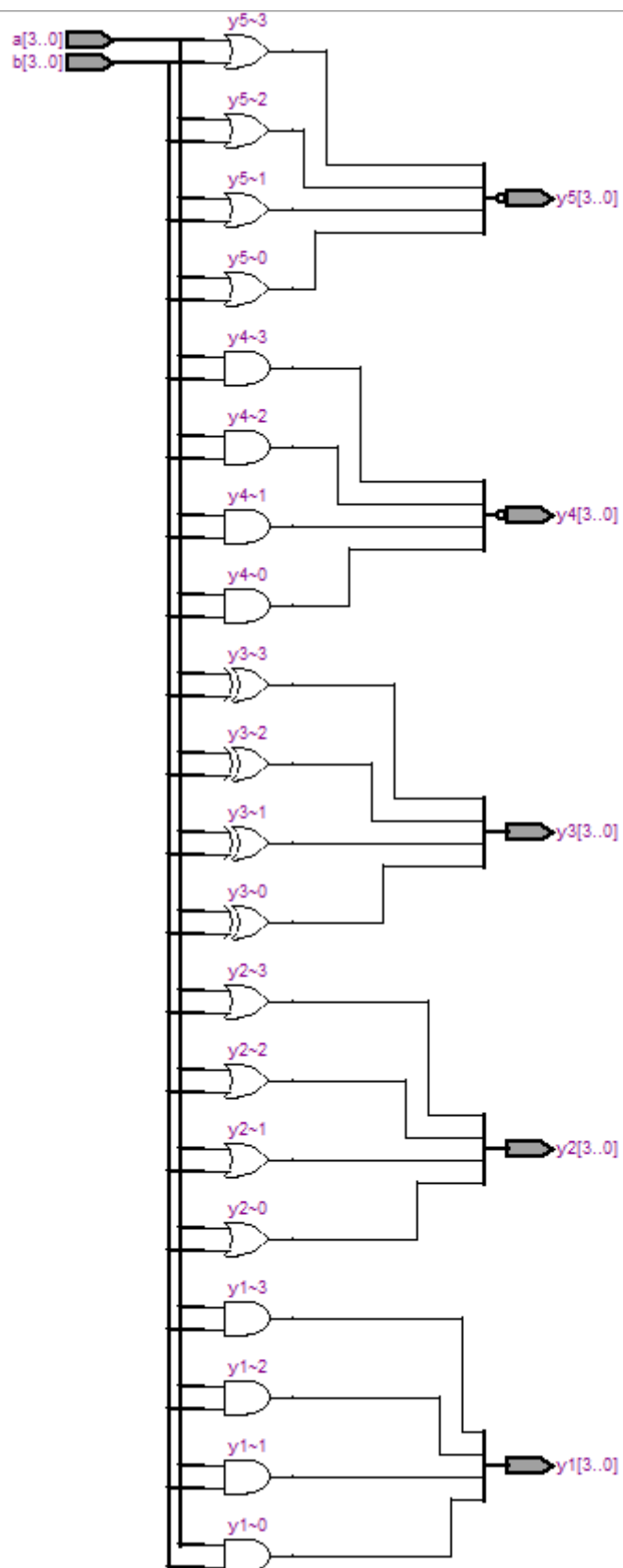
Procedimento 1

Em seguida iniciamos um outro projeto agora referente a questão 2 da preparação, que chamamos de gate. Na pasta do projeto novo foi adicionado um arquivo .vhd com as informações necessárias para a simulação proposta.

```
1  library IEEE; use IEEE.STD_LOGIC_1164.all;
2
3  entity gates is
4  port (a, b: in STD_LOGIC_VECTOR(3 downto 0);
5        y1, y2, y3, y4, y5: out STD_LOGIC_VECTOR(3 downto 0));
6  end;
7
8  architecture synth of gates is
9  begin
10     y1 <= a and b;
11     y2 <= a or b;
12     y3 <= a xor b;
13     y4 <= a nand b;
14     y5 <= a nor b;
15 end;
```

Código .VHD com informações da questão proposta.

Para adicionar esse arquivo de simulação no Quartus II basta ir em New file → VHDL, compilamos o código e corrigimos os erros (caso aconteça), então selecionamos Tools → Netlist Viewers → RTL Viewer e aí podemos conferir o circuito formado.



Circuito obtido com a simulação do código VHD.

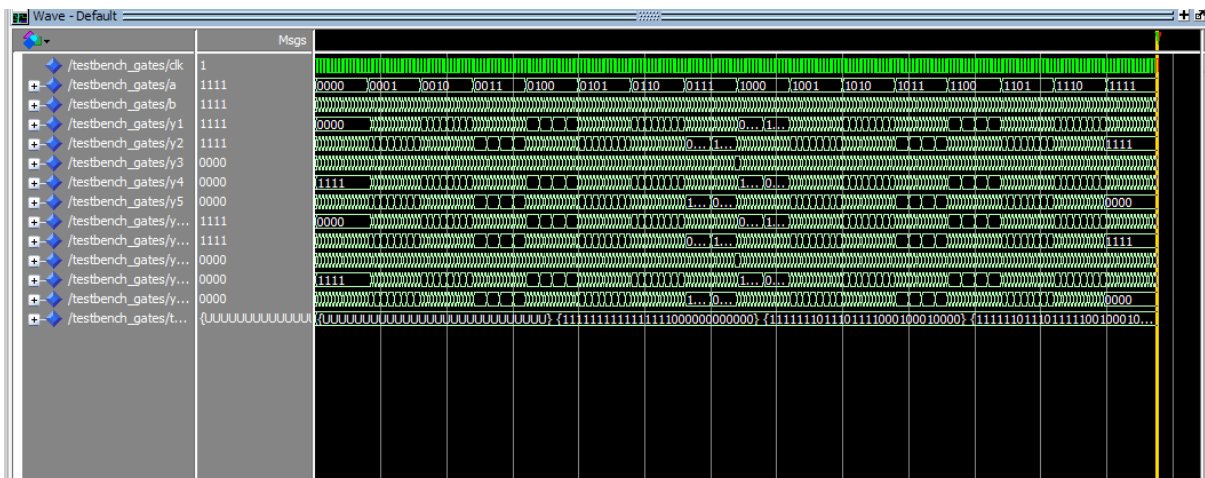
Procedimento 2

Fizemos, então, as simulação com o ModelSim. O arquivo .tv de 4 bits com a tabela verdade do circuito a ser simulado e o test bench foi nos dado pronto e então só precisamos colocá-los nas pastas corretas (.tv na pasta de simulação do projeto e o .vhd testbench na pasta de testbench).

```
00000000_00000000000011111111
00000001_00000001000111111110
00000010_000000010001011111101
00000011_000000011001111111100
00000100_000001000100111111011
00000101_000001010101111111010
00000110_000001100110111111001
00000111_00000110111111111000
00001000_00001000100011110111
00001001_00001001100111110110
00001010_00001010101011110101
00001011_00001011101111110100
00001100_00001100110011110011
00001101_00001101110111110010
00001110_00001110111011110001
00001111_00001111111111110000
```

Pequena parte da tabela verdade em 4 bits que foi fornecida

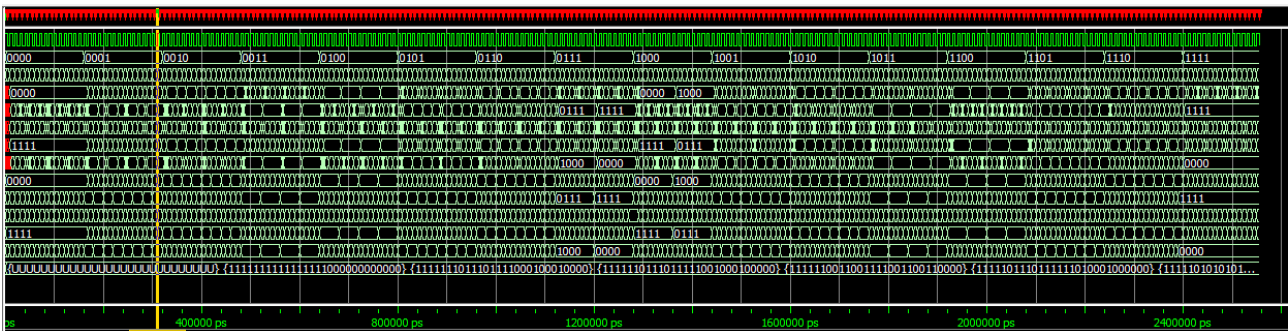
Para simularmos o RTL level selecionamos Project → Settings → EDA Tools Settings → Simulation



Resultado da simulação RTL level (sem zoom).

Vemos que não obtivemos erros com essa simulação.

Simulamos, então, o Gate level em Run EDA Simulation Tool → Gate Level e obtivemos o seguinte:



Simulação Gate Level com muitos erros (sem zoom)

```

Transcript
# Time: 2545 ns Iteration: 1 Instance: /testbench_gates
# ** Error: Vetor deu erro n. Teste: 254. Esperado yesp = '0' Valor Obtido: y3(1) = '1'
# Time: 2545 ns Iteration: 1 Instance: /testbench_gates
# ** Error: Vetor deu erro n. Teste: 254. Esperado yesp = '0' Valor Obtido: y4(1) = '1'
# Time: 2545 ns Iteration: 1 Instance: /testbench_gates
# ** Error: Vetor deu erro n. Teste: 255. Esperado yesp = '1' Valor Obtido: y1(0) = '0'
# Time: 2555 ns Iteration: 1 Instance: /testbench_gates
# ** Error: Vetor deu erro n. Teste: 255. Esperado yesp = '0' Valor Obtido: y3(0) = '1'
# Time: 2555 ns Iteration: 1 Instance: /testbench_gates
# ** Error: Vetor deu erro n. Teste: 255. Esperado yesp = '0' Valor Obtido: y4(0) = '1'
# Time: 2555 ns Iteration: 1 Instance: /testbench_gates
# ** Failure: 256 tests completed, errors = 1854
# Time: 2555 ns Iteration: 1 Process: /testbench_gates/line__71 File: D:/CIRCUITOS/gates/testbench/testbench_gates.vhd
# Break in Process line__71 at D:/CIRCUITOS/gates/testbench/testbench_gates.vhd line 120
# Simulation Breakpoint: Break in Process line__71 at D:/CIRCUITOS/gates/testbench/testbench_gates.vhd line 120
# MACRO ./gates_run_msim_gate_vhdl.do PAUSED at line 17

VSIM(paused)>

```

Transcript da simulação Gate level com 1854 erros

Vemos que obtivemos muitos erros com o gate level, isso porque na simulação Gate level se considera o atraso das portas do circuito, então fomos ao arquivo .vhd do circuito e tentamos ajustar os atrasos...

```

26 process begin
27     clk <= '1'; wait for 5 ns;
28     clk <= '0'; wait for 5 ns;
29 end process;

```

Atraso das portas antes dos ajustes

A primeira tentativa foi aumentar para 10 ns o clock em 1...

```

26 process begin
27     clk <= '1'; wait for 10 ns;
28     clk <= '0'; wait for 5 ns;
29 end process;

```

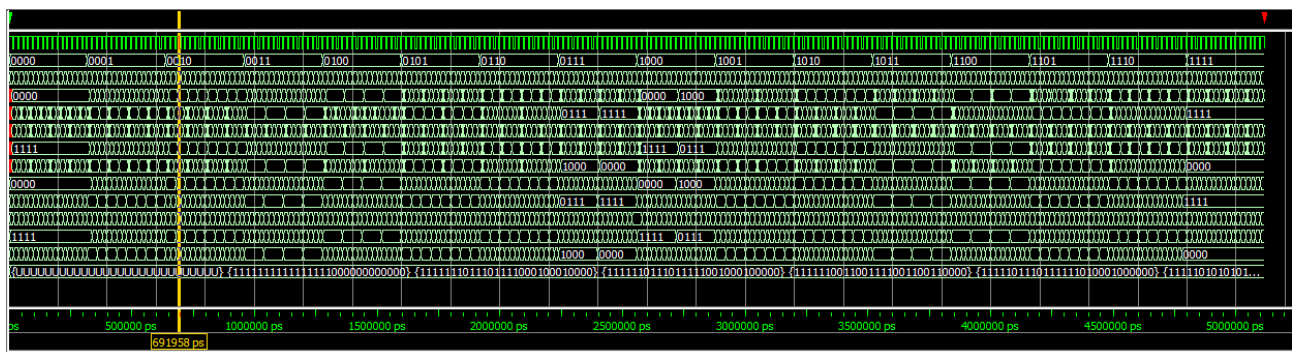

Simulamos de novo o Gate level e vemos que ainda temos erros, porém menos do que antes...

```
Transcript
# Time: 3805 ns Iteration: 1 Instance: /testbench_gates
# ** Error: Vetor deu erro n. Teste: 254. Esperado yesp = '0' Valor Obtido: y1(0) = '1'
# Time: 3820 ns Iteration: 1 Instance: /testbench_gates
# ** Error: Vetor deu erro n. Teste: 254. Esperado yesp = '1' Valor Obtido: y3(0) = '0'
# Time: 3820 ns Iteration: 1 Instance: /testbench_gates
# ** Error: Vetor deu erro n. Teste: 255. Esperado yesp = '1' Valor Obtido: y1(0) = '0'
# Time: 3835 ns Iteration: 1 Instance: /testbench_gates
# ** Error: Vetor deu erro n. Teste: 255. Esperado yesp = '0' Valor Obtido: y3(0) = '1'
# Time: 3835 ns Iteration: 1 Instance: /testbench_gates
# ** Failure: 256 tests completed, errors = 1096
# Time: 3835 ns Iteration: 1 Process: /testbench_gates/line_71 File: D:/CIRCUITOS/gates/testbench/testbench_gates.vhd
# Break in Process line_71 at D:/CIRCUITOS/gates/testbench/testbench_gates.vhd line 120
# Simulation Breakpoint: Break in Process line_71 at D:/CIRCUITOS/gates/testbench/testbench_gates.vhd line 120
# MACRO ./gates_run_msim_gate_vhdl.do PAUSED at line 17

VSIM(paused)>
```

Simulação Gate level agora com 1096 erros

Então aumentamos ainda mais o clock em 1, desta vez para 15ns...



Simulação Gate level com clock em 1 com atraso de 15ns

```
Transcript
# ** Note: (vsim-3587) SDF Backannotation Successfully Completed.
# Time: 0 ps Iteration: 0 Instance: /testbench_gates File: D:/CIRCUITOS/gates/testbench/testbench_gates.vhd
#
# add wave *
# view structure
# .main_pane.structure.interior.cs.body.struct
# view signals
# .main_pane.objects.interior.cs.body.tree
# run -all
# ** Failure: Just kidding --256 tests completed successfully.
# Time: 5115 ns Iteration: 1 Process: /testbench_gates/line_71 File: D:/CIRCUITOS/gates/testbench/testbench_gates.vhd
# Break in Process line_71 at D:/CIRCUITOS/gates/testbench/testbench_gates.vhd line 115
# Simulation Breakpoint: Break in Process line_71 at D:/CIRCUITOS/gates/testbench/testbench_gates.vhd line 115
# MACRO ./gates_run_msim_gate_vhdl.do PAUSED at line 17

VSIM(paused)>
```

Transcript da simulação provando que o Gate level agora está sem erros

Visualização em lista:

List - Default													
ps	delta	/testbench_gates/a /testbench_gates/y4/testbench_gates/yexpected4 /testbench_gates/b /testbench_gates/y5/testbench_gates/yexpected5 /testbench_gates/y1/testbench_gates/yexpected1 /testbench_gates/y2/testbench_gates/yexpected2 /testbench_gates/y3/testbench_gates/yexpected3											
3619	+1	0000	0000	XXUU	UXUU	XXXU	XXUX	UXUU	0000	0000	0000	1111	1111
3639	+1	0000	0000	XXUU	UXUU	XXXU	XXUX	UXUU	0000	0000	0000	1111	1111
3658	+1	0000	0000	XXXU	UXUU	XXXU	XXUX	UXUU	0000	0000	0000	1111	1111
3665	+1	0000	0000	XXXU	UXUU	XXXU	XXXX	UXUU	0000	0000	0000	1111	1111
4116	+1	0000	0000	XXXU	UXUU	XXXX	XXXX	UXUU	0000	0000	0000	1111	1111
4327	+1	0000	0000	XXXU	XXXU	XXXX	XXXX	XXXU	0000	0000	0000	1111	1111
4338	+1	0000	0000	XXXX	XXXU	XXXX	XXXX	XXXU	0000	0000	0000	1111	1111
4527	+1	0000	0000	XXXX	XXXU	XXXX	XXXX	XXXX	0000	0000	0000	1111	1111
5587	+1	0000	0000	X0XX	XXXU	XXXX	X1XX	XXXX	0000	0000	0000	1111	1111
5790	+1	0000	0000	X0XX	XXXU	X0XX	X1XX	XXXX	0000	0000	0000	1111	1111
5924	+1	0000	0000	X0XX	XXXU	X0XX	11XX	XXXX	0000	0000	0000	1111	1111
5962	+1	0000	0000	X0XX	XXXX	X0XX	11XX	XXXX	0000	0000	0000	1111	1111
5992	+1	0000	0000	X0XX	X0XX	X0XX	11XX	X1XX	0000	0000	0000	1111	1111
6137	+1	0000	0000	X0XX	X0XX	00XX	11XX	X1XX	0000	0000	0000	1111	1111
6146	+1	0000	0000	00XX	X0XX	00XX	11XX	X1XX	0000	0000	0000	1111	1111
6876	+1	0000	0000	00XX	00XX	00XX	11XX	11XX	0000	0000	0000	1111	1111
9332	+1	0000	0000	00XX	00XX	00XX	11X1	11XX	0000	0000	0000	1111	1111
9773	+1	0000	0000	000X	00XX	00XX	11X1	11XX	0000	0000	0000	1111	1111
9780	+1	0000	0000	000X	00XX	00XX	1111	11XX	0000	0000	0000	1111	1111
9831	+1	0000	0000	000X	00XX	000X	1111	11XX	0000	0000	0000	1111	1111